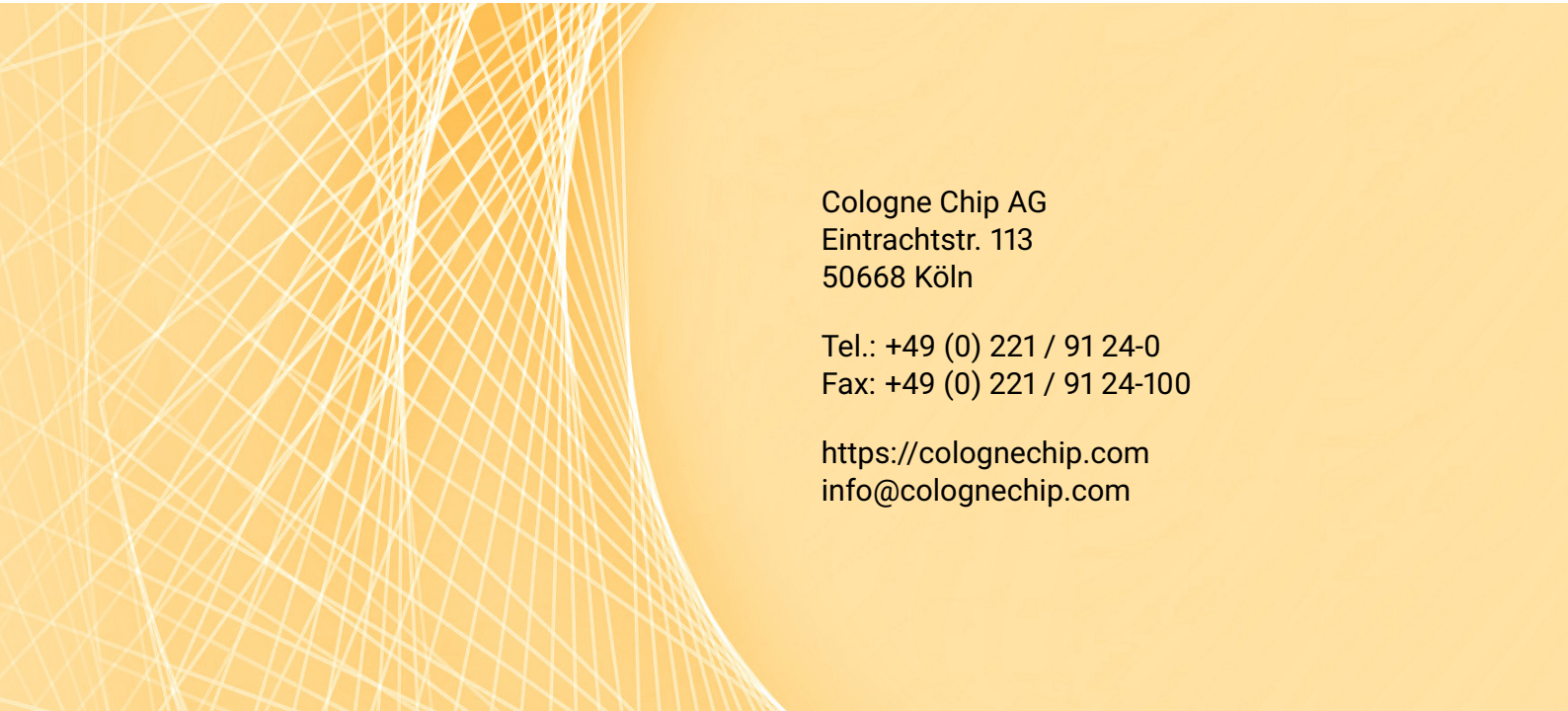# GateMate™ FPGA
Programmer Datasheet

Programmer Version 1.3

Cologne Chip AG
Eintrachtstr. 113
50668 Köln

Tel.: +49 (0) 221 / 91 24-0
Fax: +49 (0) 221 / 91 24-100

https://colognechip.com
info@colognechip.com

Contents

About this Document

This datasheet covers the GateMate^{TM} FPGA Programmer of Cologne Chip and is part of the GateMate^{TM} FPGA documentation collection.

For more information please refer to the following documents:

- Technology Brief of GateMate^{TM} FPGA ⬀
- **DS1001** – GateMate^{TM} FPGA CCGM1A1 Datasheet ⬀
- **UG1002** – GateMate^{TM} FPGA Toolchain Installation User Guide ⬀

Cologne Chip provides a comprehensive technical support. Please visit our website for more information or contact our support team.

Revision History

This Datasheet is constantly updated. The latest version of the document can be found following the link below:

> **DS1002** – GateMate^{TM} FPGA Programmer Board Ver. 1.3 Datasheet ⬀

Date	Remarks
March 2022	Initial release.

Chapter 1

Introduction

Figure 1.1: *Top view of the GateMate^{TM} FPGA Programmer (including module cover)*

The GateMate^{TM} FPGA Programmer allows an easy and flexible access to the GateMate^{TM} FPGA. The configuration can be loaded from any host computer via USB interface into the FPGA. Furthermore, the flash memory, which is typically available at the user application can be accessed as well.

The main features of the programmer are

- FTDI FT232HQ USB bridge
- USB 2.0 to JTAG and SPI bridge provided by FTDI's Multi-Protocol Synchronous Serial Engine (MPSSE)
- Mini-USB B connector
- USB-powered
- Configuration voltage VDD_CFG required from user application
- 4 status LEDs
- Form factor 35×55 mm

This document describes how to set up the GateMate^{TM} FPGA Programmer and explains the different use cases.
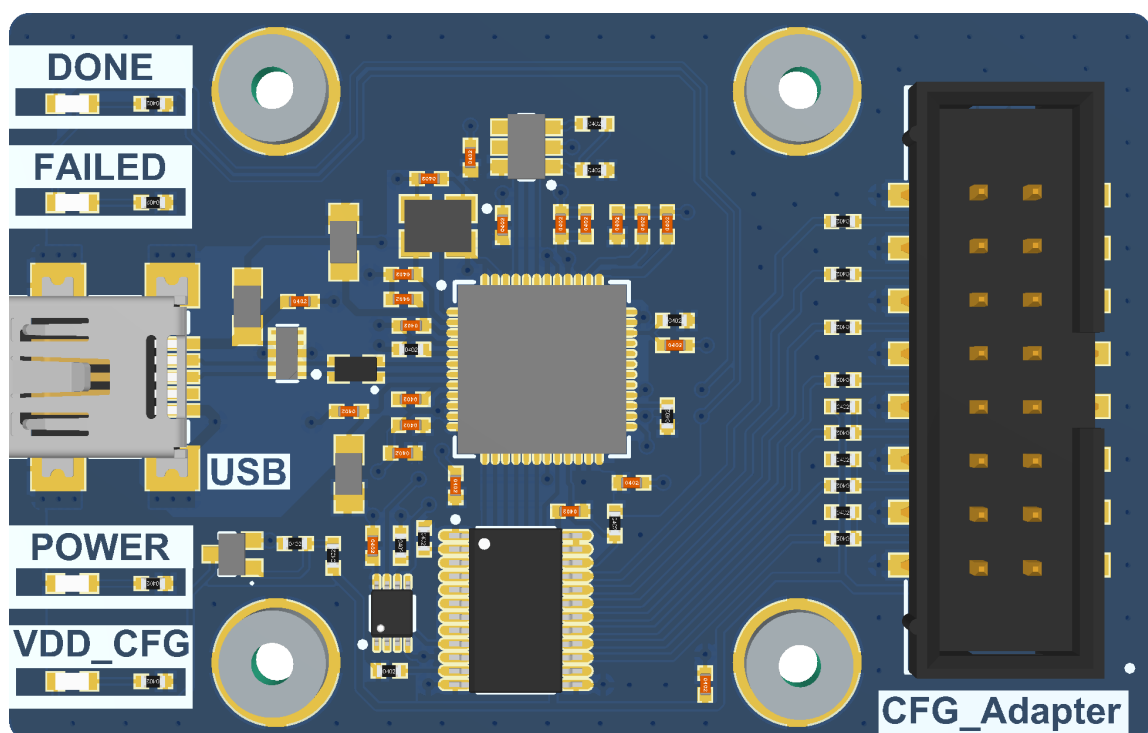
Figure 1.2: *Top view of the GateMate^{TM} FPGA Programmer (module cover removed)*

Chapter 2

Block Diagram

Figure 2.1 shows the block diagram of the GateMate^{TM} FPGA Programmer and its embedding between the host controller and the user application.

Typically, the host controller is any computer that comes with an USB interface. The programmer software runs on both `Microsoft Windows` and `Linux`.

Four LEDs display the current state of the GateMate^{TM} FPGA Programmer device as described in Table 2.1.

Table 2.1: *GateMate^{TM} FPGA Programmer LED signaling*

LED	color	Description
D1	green	USB power available from host controller
D3	green	`Configuration Done` signaling
D4	red	`Configuration Failed` signaling
D5	green	Power from FPGA application (configuration GPIO bank) available

The location of the main components can be found in Figure 2.2. Table 2.2 gives a short specification of these components. For more details, please see the schematics in Appendix A.
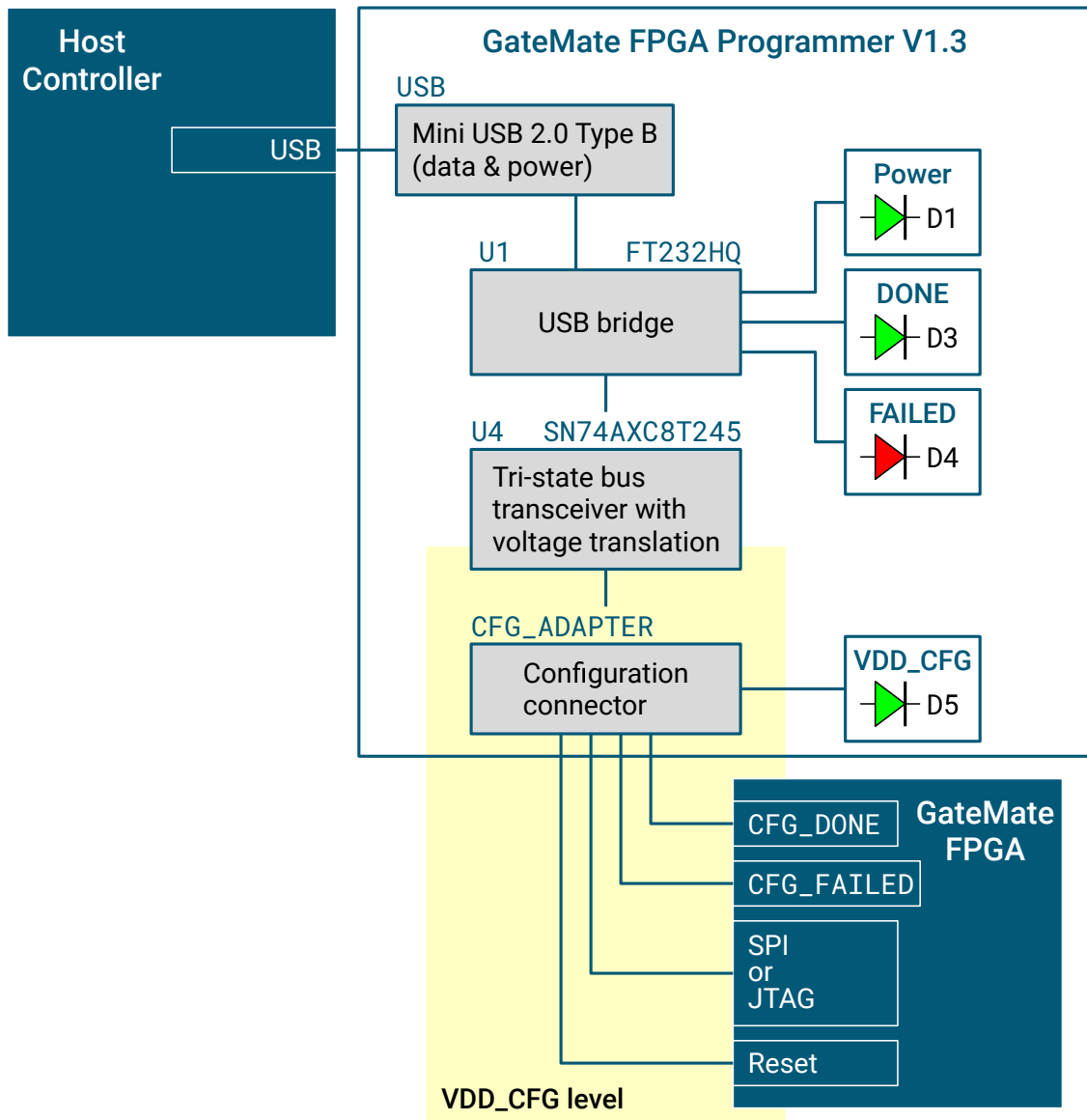
Figure 2.1: *Block diagram of the GateMate^{TM} FPGA Programmer*
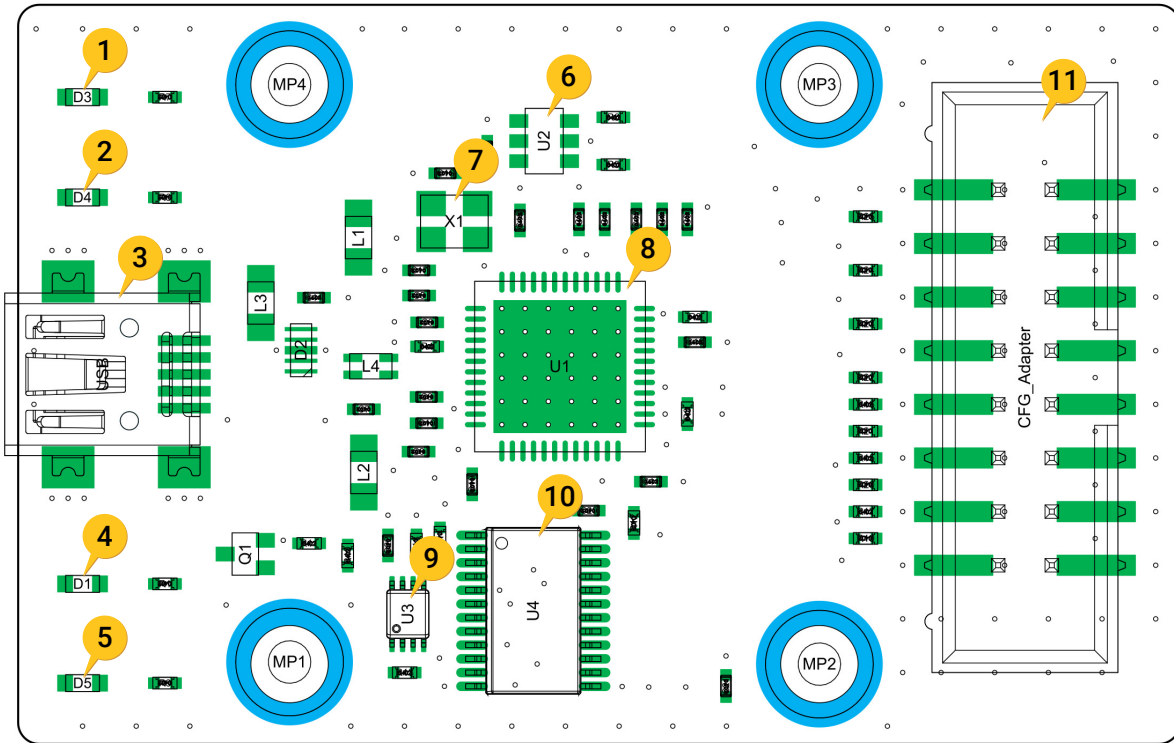
Figure 2.2: *Main components of the GateMate^{TM} FPGA Programmer (top side)*

Table 2.2: *Main components of the GateMate^{TM} FPGA Programmer (top side)*

Number	Designator	Component
1	D3	Status LED `CFG_DONE` (from USB bridge)
2	D4	Status LED `CFG_FAILED_N` (from USB bridge)
3	USB	Mini USB 2.0 Type B Receptacle
4	D1	Status LED for power from USB host
5	D5	Status LED for GPIO power from CFG_Adapter
6	U2	Microchip 93LC56BT-I/OT, 2 Kbit 128×16, 2.5 V, FTDI configuration EEPROM
7	X1	Oscillator 12.000 MHz ±30ppm, 8 pF
8	U1	Microchip FTDI FT232HQ-Tray
9	U3	Microchip SN74AUP2G00DCUR, failed detection latch
10	U4	Microchip SN74AXC8T245PWR, bus transceiver U1 / CFG_Adapter
11	CFG_Adapter	Male box header, pitch 2.54 mm, 2×8 pins

Chapter 3

Startup

3.1 Connection to the User Application

To connect the GateMate^{TM} FPGA Programmer to the user application, the connector CFG_Adapter must be used. Figure 3.1 shows the signals of this connector. Depending on the user's requirements, it can act as JTAG or SPI bus. The detailed BGA connection is described in Table 3.1 for both interfaces.
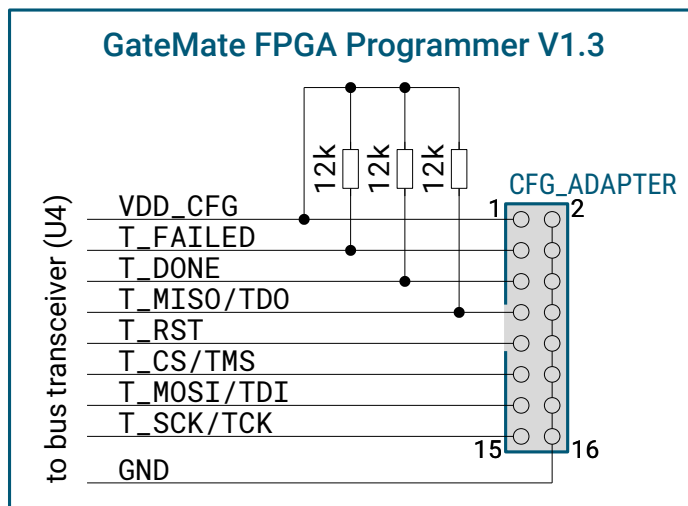

Figure 3.1: *Connecting the GateMate^{TM} FPGA Programmer with the GateMate FPGA*

The GateMate^{TM} FPGA Programmer is USB bus powered. The connector CFG_Adapter requires power supply from the FPGA configuration GPIO bank (signal VDD_CFG) for opera-

tion. The pull-up resistors shown in Figure 3.1 are populated on the programmer's printed circuit board (PCB).

No further hardware setup is required.

Table 3.1: *GateMate^{TM} FPGA Programmer configuration adapter*

GateMate^{TM} FPGA Programmer CFG_ADAPTER		GateMate FPGA JTAG interface		GateMate FPGA SPI interface	
pin	signal name	ball	signal name	ball	signal name
1	VDD_CFG	U3[1]	VDD_WA	U3[1]	VDD_WA
3	T_FAILED	V2	CFG_FAILED_N	V2	CFG_FAILED_N
5	T_DONE	V3	CFG_DONE	V3	CFG_DONE
7	T_MISO / TDO	U2	JTAG_TDO	P1	SPI_D1
9	T_RST	T15	RST_N	T15	RST_N
11	T_CS / TMS	T3	JTAG_TMS	N3	SPI_CS_N
13	T_MOSI / TDI	T2	JTAG_TDI	P2	SPI_D0
15	T_SCK / TCK	R3	JTAG_TCK	N4	SPI_CLK
2,4,6,..,16	GND	V18[1]	GND	V18[1]	GND

[1] and others

3.2 Connection to the Host Controller

The GateMate^{TM} FPGA Programmer requires a computer with Linux or Windows operating system as follows:

- Supported Linux environments:
    - Debian-based Linux (Debian, Ubuntu, ...) with `apt` package manager
    - Arch-based Linux (Arch, Manjaro, ...) with `pacman` package manager
    - Red Hat-based Linux (Fedora, ...) with `dnf` or `yum` package manager

- Windows environments:
    - Windows 7 or later, 64 bit
    - Zadig USB driver installer
      `https://zadig.akeo.ie/`

When first plugged into the computer's USB port, drivers should load by default.

In Windows environments, however, it is necessary to change the default USB driver using Zadig ⤢. Download the software and connect the GateMate^{TM} FPGA Progammer to any USB port. In the Zadig Window, select **Options > List All Devices** to refresh the device list. From the drop-down list, select **GateMate FPGA Programmer 1.3A**. Now select **libusb-win32 (any version)** from the driver list and replace the drivers (see Figure 3.2).
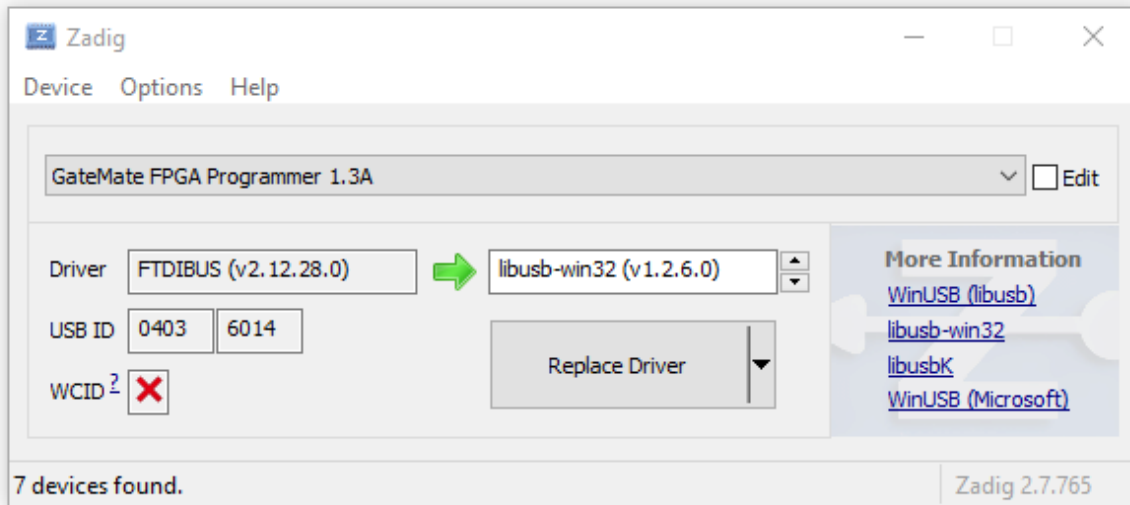
Figure 3.2: *Zadig Window with selected GateMate^{TM} FPGA Programmer*

Replacing drivers might take a moment. Your GateMate^{TM} FPGA Programmer should then be listed as **libusb-win32** device in the Device Manager as shown in Figure 3.3.

In Linux environments, no driver installation is required. The device should enumerate as FT232H using the `lsusb` command.
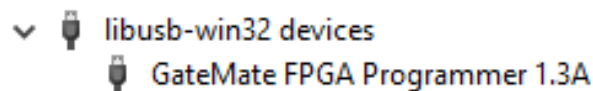
Figure 3.3: *GateMate^{TM} FPGA Programmer in Device Manager*

The tool setup for the Cologne Chip GateMate^{TM} series is described in the user guide

UG1002 – GateMate^{TM} FPGA Toolchain Installation User Guide ⬀

Chapter 4

Loading the FPGA Configuration

GateMate FPGA configuration can be loaded either via the JTAG or SPI interfaces. Both options are described in this chapter.

It is assumed that the user application has, among other things, a GateMate FPGA and a flash memory from which the configuration is loaded during normal operation. In this chapter it is described, how configuration bit files can be handled alternatively with the GateMate^{TM} FPGA Programmer during the application development.

4.1 Loading the FPGA Configuration via the JTAG Interface

In this mode, the programmer is configured to send the configuration bitstream directly to the FPGA via JTAG as shown in Figure 4.1. The FPGA configuration mode pins `CFG_MD[3:0]` must be set to `0x C` (JTAG).

An active `CFG_DONE` and inactive `CFG_FAILED` signal indicates successfull configuration. The `CFG_FAILED` signal may send a pulse during configuration in case of any error. Note that the `CFG_DONE` and `CFG_FAILED` pins can be configured as user GPIO. In this case no configuration status can be read back.

4.2 Loading the FPGA Configuration via the SPI Interface

In this mode, the programmer is configured to send the configuration bitstream directly to the FPGA via SPI as shown in Figure 4.2. The FPGA configuration mode pins `CFG_MD[3:0]` must be set to `0x 4` (SPI slave mode).
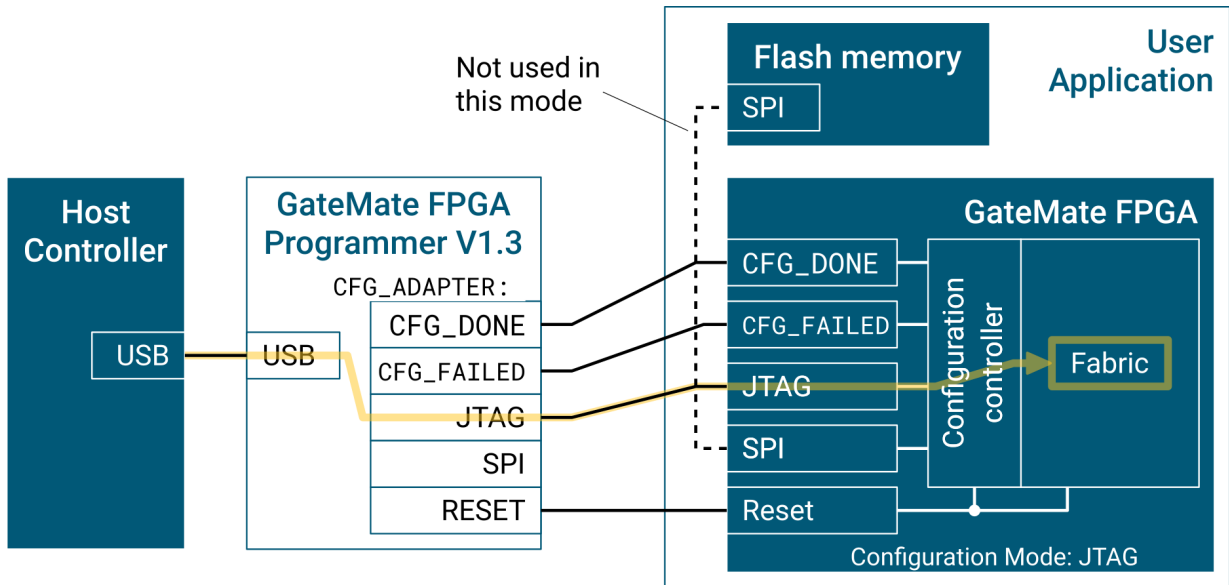
Figure 4.1: *Loading the FPGA configuration via the JTAG interface*

An active `CFG_DONE` and inactive `CFG_FAILED` signal indicates successfull configuration. The `CFG_FAILED` signal may send a pulse during configuration in case of any error. Note that the `CFG_DONE` and `CFG_FAILED` pins can be configured as user GPIO. In this case no configuration status can be read back.
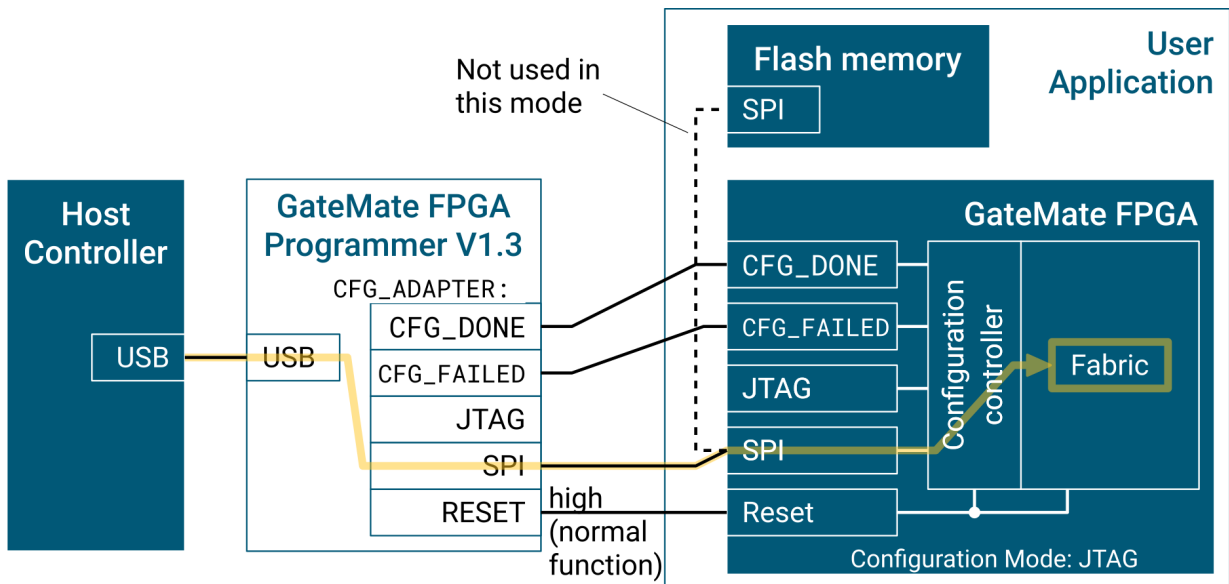
Figure 4.2: *Loading the FPGA configuration via the SPI interface*

Chapter 5

Access to the Flash Memory

The GateMate^{TM} FPGA Programmer offers an easy way to access the user application's flash memory. Again, two ways are possible as described in this chapter.

5.1 Access to the Flash Memory via the JTAG Interface

The GateMate^{TM} FPGA configuration controller has a built-in JTAG-SPI-bridge to access an external SPI flash memory using the JTAG interface.

Handling of the JTAG-SPI-bridge is described in the GateMate^{TM} FPGA datasheet ☐.

In this mode, the programmer is configured to send the configuration bitstream or any other data to the external SPI flash memory via JTAG as shown in Figure 5.1. The FPGA configuration mode pins `CFG_MD[3:0]` must be set to `0x C` (JTAG). Further prerequisite is that the FPGA configuration bank is not configured as user GPIO.

In this configuration, both `CFG_DONE` and `CFG_FAILED` signals don't output any status information.

5.2 Access to the Flash Memory via the SPI Interface

The following configuration requires a direct connection of the SPI signals to the external flash memory. Both external flash memory and GateMate^{TM} FPGA may share the same signals.
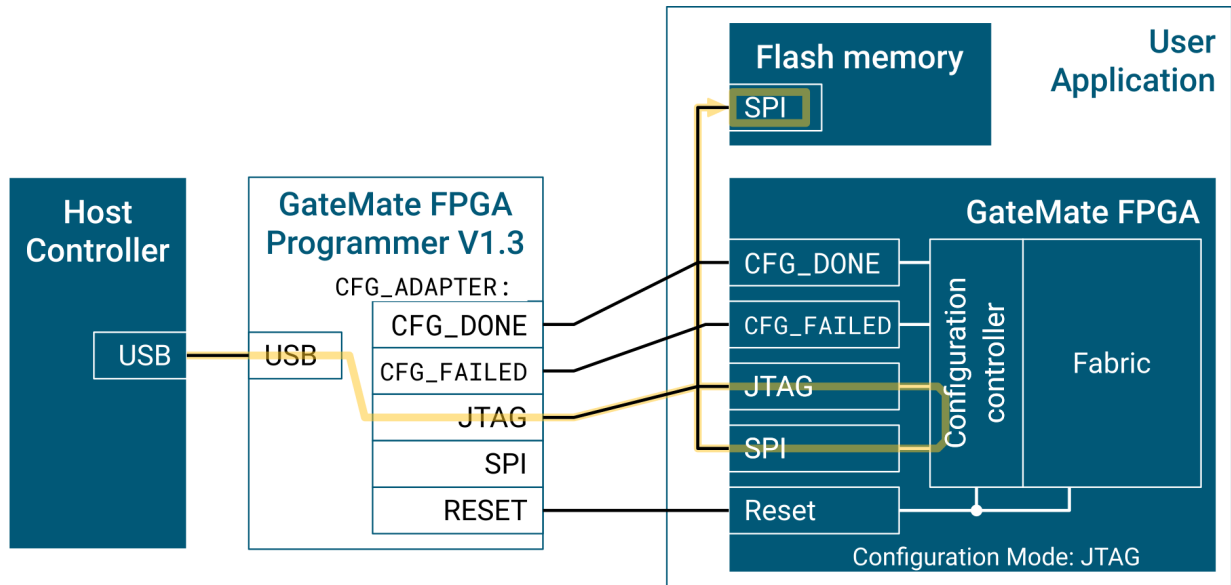
Figure 5.1: *Access to the flash memory via the JTAG interface*

In this mode, the programmer is configured to send the configuration bitstream or any other data directly to the external SPI flash memory as shown in Figure 5.2. Prerequisite is that the FPGA acts passively on the SPI bus. If both FPGA and external flash memory share the same bus signals, it is required to keep the FPGA in active reset while programming the external flash memory.

After successfull programming, the reset signal may be released to start the FPGA configuration e.g. in SPI master mode.
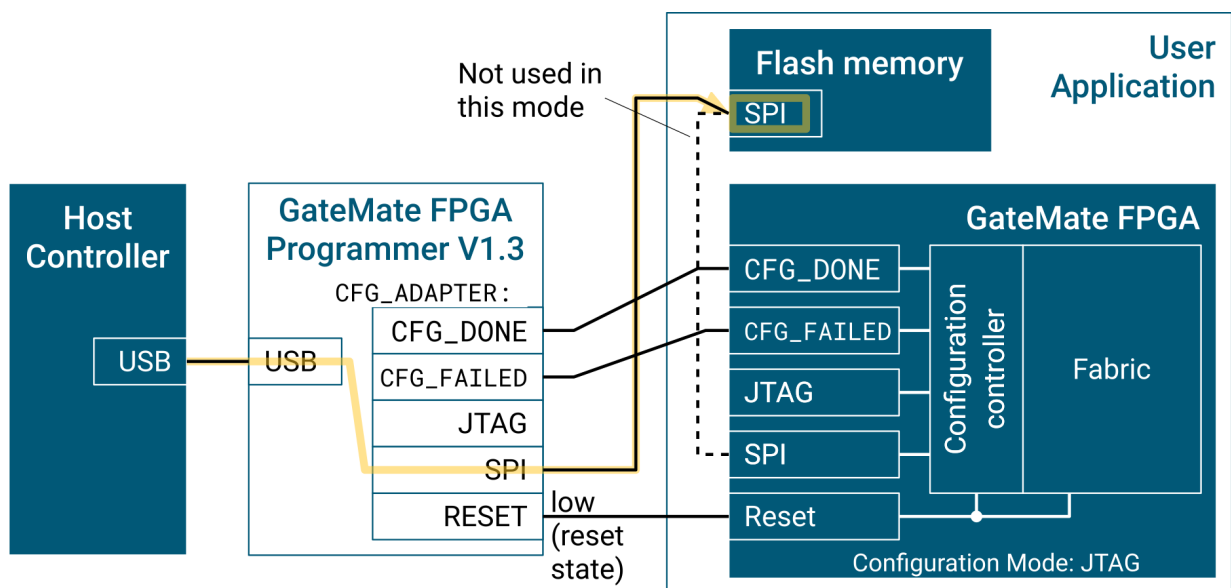
Figure 5.2: *Access to the flash memory via the SPI interface*

Chapter 6

Electrical Characteristics

Table 6.1: *Absolute maximum characteristics of the GateMate^{TM} FPGA Programmer*

Symbol	Min	Typ	Max	Unit	Description
	−40		125	°C	Junction temperature
V_{BUS}	−0.3		5.8	V	USB Voltage
VDD_{CFG}	−0.5		4.2	V	Target Supply Voltage
V_i	−0.5		4.2	V	Input Signals on `CFG_ADAPTER`
V_o	−0.5		4.2	V	Output Signals on `CFG_ADAPTER`
T_{stg}	−55		150	°C	Storage temperature

Table 6.2: *Operating characteristics of the GateMate^{TM} FPGA Programmer*

Symbol	Min	Typ	Max	Unit	Description
	−40		85	°C	Operating temperature
V_{BUS}	4.75	5.0	5.25	V	USB Voltage
VDD_{CFG}	1.4		2.75	V	Target Supply Voltage
V_i	0		2.75	V	Input Signals
V_o	0		2.75	V	Output Signals Active
	0		VDD_{CFG}	V	Output Signals Tri−State

Chapter 7

Mechanical Dimensions
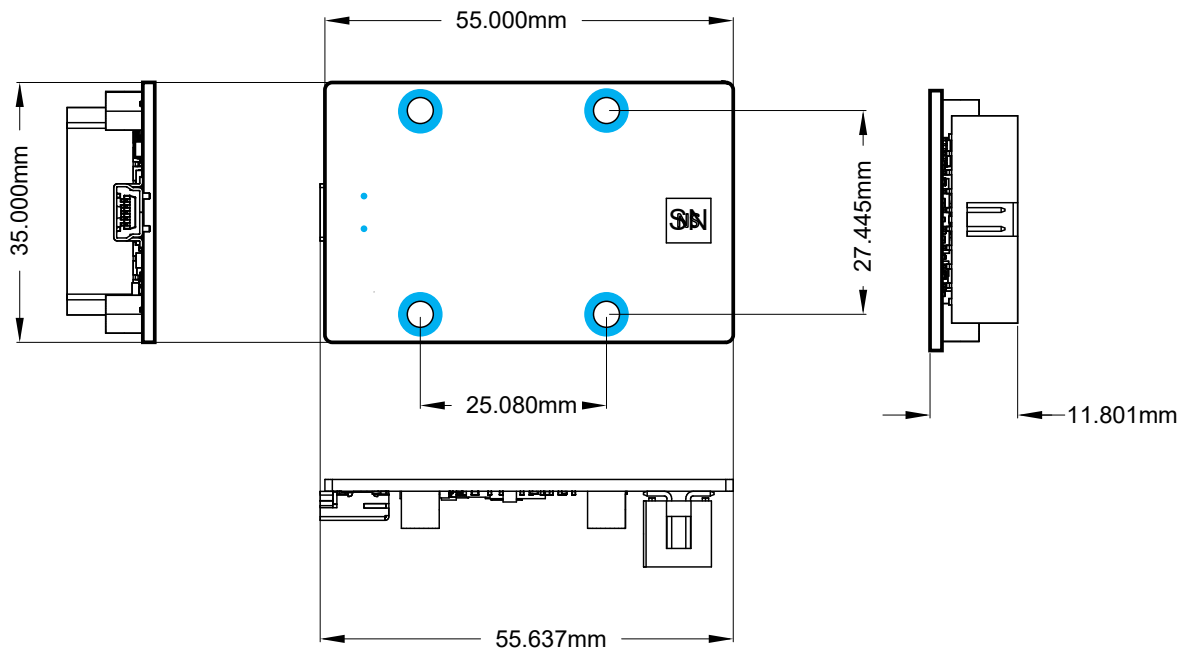
Figure 7.1: *Mechanical dimensions of the GateMate^{TM} FPGA Programmer (top side)*

Appendix A

Programmer Schematics

Programmer Schematics

CCGM1A1

Programmer (FTDI 232H)

Project:	GateMate FPGA-Programmer		Revision:	1.3A
Title:	Programmer Wrapper (FTDI 232H)			
Size: A4	Author: CCAG	Date: 23 Mar 2022	Sheet 1 of 2	
File:	FTDI_Wrapper_Programmer.SchDoc			

Cologne Chip AG
Eintrachtstr. 113
50668 Cologne
Germany

Power LED

CFG_DONE

CFG_FAILED

VDD_CFG
min. 1.4V
max. 2.5V

VCCA
C3 0.1u 6.3V X5R
GND

VCORE
C4 0.1u 6.3V X5R
GND

VCC3V3
L1 MI0805K601R-10 600R 1,5A, 0,1R
C5 4.7u 6.3V X5R

VPHY
C6 0.1u 6.3V X5R
GND

VCC3V3
L2 MI0805K601R-10 600R 1,5A, 0,1R
C7 4.7u 6.3V X5R

VPLL
C8 0.1u 6.3V X5R
GND

VCC3V3
C9 4.7u 6.3V X5R
C10 0.1u 6.3V X5R
GND

VCC3V3
C11 0.1u 6.3V X5R
C12 0.1u 6.3V X5R
C13 0.1u 6.3V X5R
GND

VCC3V3
C15 0.1u 6.3V X5R
R5 12k
U2
6 VCC
VSS 2
GND
EECS 5 CS
EECLK 4 CLK
EEDATA 3 DI
DO 1
R6 2k 1%
93LC56BT-I/OT

VPHY VPLL VCC3V3
VCC3V3

U1 FT232HQ-Tray

VBUS
40 VREGIN
VPHY 3
VPLL 8
VCCIO 12
VCCIO 24
VCCIO 46

VCC3V3 39 VCCD
VCORE 38 VCORE
VCCA 37 VCCA

D- LF 6 DM   D- LF
D+ LF 7 DP   D+ LF

VCC3V3 R2 12k 34 RESET#
5 REF

EECS 45 EECS
EECLK 44 EECLK
EEDATA 43 EEDATA

1 OSCI
X1
C16 8p 6.3V X5R
1 GND GND 3
2 GND GND 4
7M-12.000MAHV-T
C17 8p 6.3V X5R
R4 12k
C18 10n 6.3V X5R
2 OSCO
42 TEST

ADBUS0 13 SCK/TCK
ADBUS1 14 MOSI/TDI
ADBUS2 15 MISO/TDO
ADBUS3 16 CS/TMS
ADBUS4 17 RST
ADBUS5 18 DONE
ADBUS6 19 FAILED long
ADBUS7 20 T_POWERED

ACBUS0 21 OE
ACBUS1 22 FF_RESET
ACBUS2 26 LED_GPIO1
ACBUS3 28 DETECT
ACBUS4 29 LED_GPIO2
ACBUS5 30 AC5 AC6
ACBUS6 31 AC7
ACBUS7 32 AC8
ACBUS8 33 AC9
ACBUS9

AGND AGND AGND GND GND GND GND GND GND GND GND GND GND GND GND
4 9 41 10 11 22 23 25 35 36 47 48 49

GND

Project:	GateMate FPGA-Programmer	Revision:	1.3A
Title:	Programmer (FTDI 232H)		
Size: A4	Author: CCAG	Date: 23 Mar 2022	Sheet 2 of 2
File:	Programmer_FTDI.SchDoc		

Cologne Chip AG
Eintrachtstr. 113
50668 Cologne
Germany

Cologne Chip
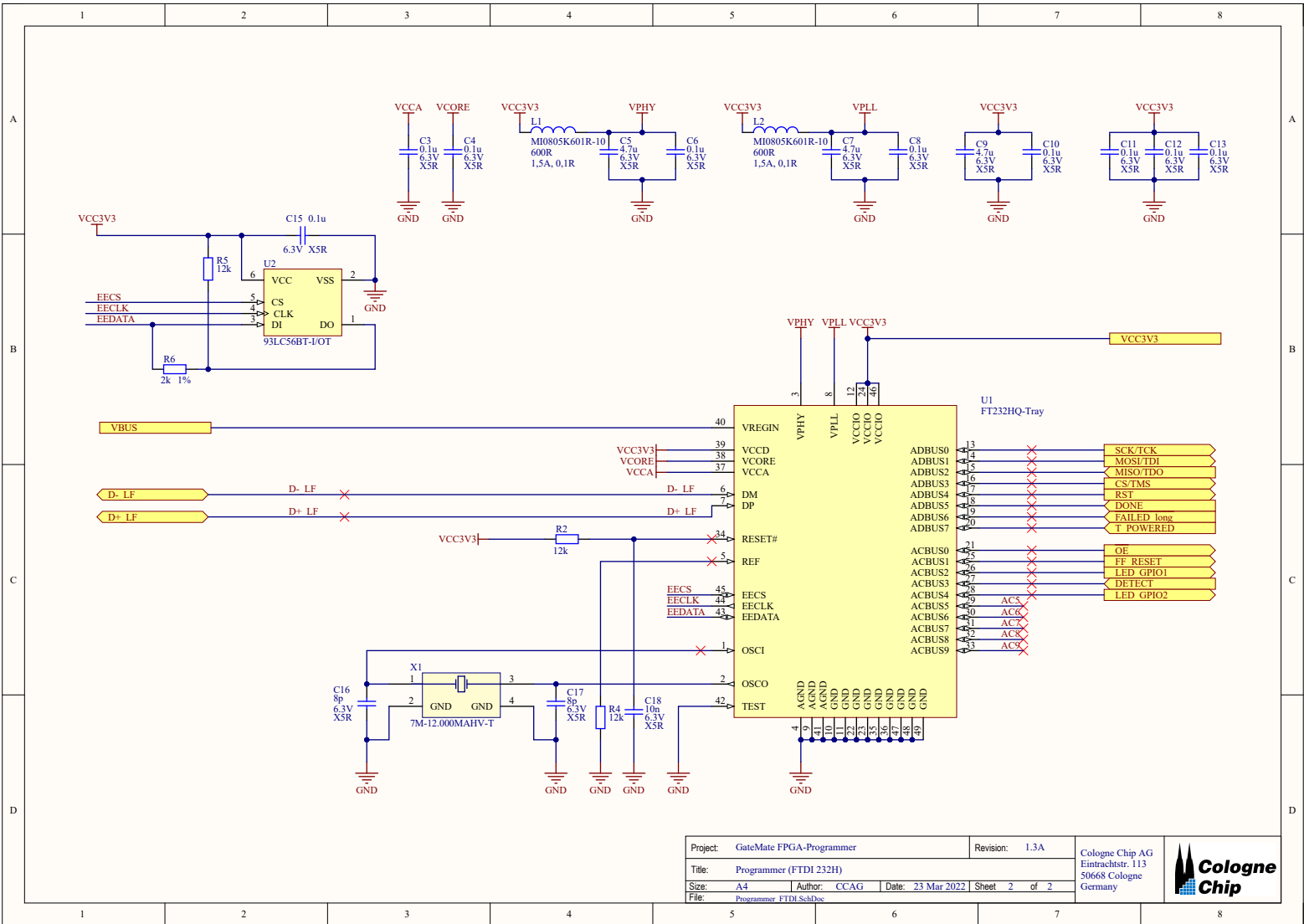
GateMate^{TM} FPGA
Programmer Datasheet
Programmer Version 1.3
DS1002
March 2022